Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Štefan Šurk

Mobilna aplikacija za upravljanje turnirjev

diplomsko delo

visokošolski strokovni študijski program prve stopnje računalništvo in informatika

Mentor: viš. pred. dr. Igor Rožanc

Ljubljana, 2016

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V diplomski nalogi predstavite razvoj mobilne aplikacije za organiziranje športnih turnirjev. V ta namen najprej proučite različne turnirske razporeditvene sisteme, identificirajte njihove ključne težave in na tej podlagi predlagajte kombiniran sistem, ki bo čimbolj korekten. Idejo realizirajte z mobilno aplikacijo, katere sistematičen razvoj in uporabo na primeru malega nogometa tudi prikažite. Nalogo zaključite z analizo uporabe aplikacije.

Izjava o avtorstvu diplomskega dela

Spodaj podpisani Štefan Šurk sem avtor diplomskega dela z naslovom:

Mobilna aplikacija za upravljanje turnirjev (angl. *Mobile application for tournament managment*)

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom viš. pred. dr. Igorja Rožanca,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 3. marca 2016 Podpis avtorja:

Zahvaljujem se svojim bližnjim, ki so mi vedno stali ob strani. Mojemu mentorju viš. pred. dr. Igorju Rožancu, katerega napotki in vodenje so bili zlata vredni. Mojemu družbenemu krogu, ki mi je dal potrebno zaupanje in motivacijo za izpeljavo študija. Hvala vsem.

Diplomsko delo posvečam očetu Viktorju,
mami Francki in moji najdražji Moniki.

Kazalo

Seznam uporabljenih kratic

kratica	**angleško**
FIFA	Fédération Internationale de Football Association
CAF	Confederation of African Football
CONMEBOL	South American Football Confederation
AFC	American Football Conference
CONCACAF	The Confederation of North, Central America and Caribbean Association Footbal
UEFA	Union of European Football Associations

Povzetek

Naslov: Mobilna aplikacija za upravljanje turnirjev

Diplomska naloga predstavlja mobilno aplikacijo za upravljanje turnirjev.

Ogledali smo si klasične razporeditvene sisteme: krožni, švicarski in izločilni sistem. Prednost krožnega sistema je pravičnost, vendar je časovno potraten. Švicarski sistem je ustrezen glede časa in pravičnosti. Njegova slabosti leži v vnaprej določenem poteku tekmovanja. Izločilni sistem je naključen in hiter a zaradi tega pade v pasti skupine smrti in lokacije ekip. Zaradi navedenih ugotovitev smo se odločili za uporabo kombiniranega sistema.

Razvili smo mobilno aplikacijo za ustvarjanje, urejanje, brisanje in pregled igralcev, ekip, sistemov in tekmovanj. Med shranjevanjem tekmovanja se opravi žreb, ki se uporabniku pregledno prikaže. Med delom smo uporabili naslednje tehnologije in orodja: platforma `Android`, programska knjižnica `SQLite`, vgrajeno razvijalno okolje `Android Studio 1.3.2`, repozitorij izvorne kode `Git`.

Med analizo realizacije smo si ogledali že obstoječe aplikacije in jih med sabo primerjali. Na obstoječih podatkih smo opravili primer uporabe.

Ključne besede: turnir, turnirski sistem, mobilna aplikacija, android.

Abstract

Title: Mobile application for tournament managment.

The thesis presents a mobile application for tournament managment.

We took a glance at some classic tournament systems: Round-Robin, Swiss and Knockout System. The Round-Robin System advantage is fairness, but it's time consuming. Swiss System is time efficient and fair. His weakness lies with it's directed competition. Knockout System is random and fast, but falls into traps of the group of death and team location. According to listed foundings we decided to use a combined system.

We developed a mobile application for creating, editing, deleting and showing players, teams, systems and tournaments. A draw is made during the saving tournament process that a user can see. During work we used the following technologies: Android platform, SQLite library, Android Studio 1.3.2 tool and Git repository.

During realization analysis we took a look at existing applications and compared them with each other. On existing data we showed a use case scenario.

Keywords: tournament, tournament system, mobile application, android.

Poglavje 1

Uvod

Živimo v času, ko ne moremo odmisliti številnih prežečih kriz, ki se pojavljajo na obzorju naše družbe. Potrebujemo simbol upanja, ki ljudi različnih družbenih slojev povezuje in jih navdaja z radostjo.

Ta simbol je pri nas že vrsto let šport, ki povezuje vse generacije na športnih prizoriščih ali vsaj pred televizijskimi sprejemniki. Vsako leto se organizira ogromno število športnih prireditev, ki privabijo množice ljudi z vsega sveta. Na večjih prireditvah se že uporabljajo ustaljene prakse organiziranja, vodenja in izvedbe prireditve. Večina rezultatov se zapisuje računalniško, kar omogoča lažji pregled, objavljanje in statistično analizo podatkov. Uporaba tehnologije ponuja organizatorjem lažjo pripravo in vodenje tekmovanja.

Na velikih tekmovanjih ne primanjkuje sredstev, ki omogočajo uporabo tovrstne tehnologije. Vendar na področju športa prevladuje množica manjših prireditev, ki so po navadi omenjene le v lokalnem časopisu. Organizacijo omenjenih prireditev večinoma prevzamejo okoliška športna društva, ki s pridobljenimi občinskimi sredstvi opravijo večino dela. Večinoma se morajo organizatorji posluževati ročnega načina organiziranja in vodenja tekem ter zapisovanja rezultatov.

Močno je izpostavljena želja po cenovno ugodnemu računalniškem sistemu, ki bi omogočal lažje vodenje manjših prireditev, od žreba tekmovanja

do objave rezultatov. Uporabniku želimo pomagati najprej pri žrebu tekmovanja. Na prireditvah se uporabljajo različni razporeditveni sistemi. Ogledati si moramo posamezne vrste sistemov in ugotoviti, katere sisteme lahko uporabimo na manjših tekmovanjih. Moramo se zavedati, da je število športov preveliko. Da bi lahko ustregli vsem, bo potrebno izbrati vzorec najprimernejših. Glede na to, da imamo izkušnje z organizacijo turnirjev v malem nogometu, bomo temelje postavili v tej kategoriji. Ogledali si bomo, kako lahko organizatorjem že med žrebom samim olajšamo delo. Razvili bomo aplikacijo za ustvarjanje turnirjev, s katero bo lahko uporabnik preprosto načrtoval naslednje tekmovanje. Za aplikacijo bomo izbrali primerno platformo in poskusili zagotoviti njeno funkcionalnost na širšem številu športov z različnimi razporeditvenimi sistemi. Na koncu bomo preverili, če smo ustvarili sistem, s katerim lahko organizatorjem olajšamo načrtovanje tekmovanj.

Poglavje 2

Turnirski razporeditveni sistemi

Turnirski razporeditveni sistem [1] je kombinacija vnaprej določenih parametrov, s katerimi razporedimo tekmovalce ali ekipe na tekmovanju. Določa lahko vrstni red nastopov, medsebojne obračune ali skupine na tekmovanju. Z izbiro razporeditvnega sistema vplivamo na uspešnost tekmovalcev ali ekipe in zanimivost celotnega tekmovanja. Dejavniki, ki vplivajo na odločitev o vrsti sistema, so po navadi čas tekmovanja, število igralcev, vrsta športa, itd.

2.1 Vrste turnirskih razporeditvenih sistemov

Na svetu obstaja več vrst razporeditvenih sistemov, ki se uporabljajo za različne namene. Zaradi splošnih dejavnikov je izbira pravega sistema zelo pomembna. Poznamo klasične sisteme, kot so krožni, švicarski in izločilni sistem. Nekateri sistemi so usmerjeni na posamezne športe, kot sta sistema za bridge in poker. Na tekmovanjih se večinoma uporabljajo kombinacije klasičnih sistemov. Poznamo tudi večstopenjske, skupinske ter različne izločilne sisteme. Večstopenjski sistemi uporabljajo ponavadi različno vrsto za skupinski del, kot za finalni del. V nadaljevanju si bomo ogledali klasične sisteme in izpostavili njihove prednosti in slabosti.

2.1.1 Krožni sistem

Krožni sistem angl. `Round Robin` [2] se po navadi uporablja za tekmovanja, ki potekajo daljše časovno obdobje. Sodelujoči tekmovalci ali ekipe morajo nastopati vsak proti vsakemu.

2.1.2 Švicarski sistem

Zaradi časovne porabe krožnega sistema se je razvila potreba po sistemu, ki bi organizatorjem prihranil čas. Leta 1895 je J. Muller razvil tako imenovani švicarski sistem angl. `Swiss System` [3], ki je pridobil ime po svoji prvi uporabi na šahovskem tekmovanju v Zürichu, Švici. Za pravilno delovanje sistema je potrebno razporediti tekmovalce ali ekipe po točkah od prvega do zadnjega. Tekmovalci ali ekipe brez točk se navede na dno liste. V prvem krogu se zgornja polovica sodelujočih pomeri s spodnjo. Po vsakem krogu se oblikujejo skupine zmagovalnih in poraženih tekmovalcev ali ekip. Na koncu zmaga tekmovalec ali ekipa, ki ostane sam/a v skupini zmagovalcev.

2.1.3 Izločilni sistem

Izločilni sistem angl. `Knockout System` [4] je časovno gledano najučinkovitejši razporeditveni sistem. Na začetku tekmovanja opravimo žreb tekmovalcev ali ekip in določimo medsebojne obračune. V naslednje kroge napredujejo le zmagovalci, ki se na podlagi izžrebane sheme pomerijo z ostalimi zmagovalci. Na koncu se v finalu pomerita preostala tekmovalca ali ekipi. Za delovanje sistema mora biti število sodelujočih potenca števila 2. V nasprotnem primeru je potrebno opraviti dodatne kvalifikacije pred tekmovanjem, da se izloči višek sodelujočih.

2.2 Sistem malega nogometa

Odločili smo se, da se bomo osredotočili na problematiko organizacije tekmovanja v malem nogometu. Po navadi se uporablja sistem, ki je kombinacija krožnega in izločilnega sistema. Najprej se sodelujoče ekipe razdeli v skupine, ki igrajo medsebojne dvoboje vsaka proti vsaki. Po skupinskem delu napreduje določeno število sodelujočih ekip v finalni del tekmovanja. V finalnem delu se ekipe pomerijo, glede na dogovorjeno shemo tekmovanja, v izločilnih dvobojih. V finalu se najboljši ekipi pomerita za zmago na tekmovanju.

Za potrebe osredotočenosti smo določili naslednje parametre, ki ponazarjajo organizacijo tipičnega tekmovanja v malem nogometu:

- 4 + 1 sistem igranja[1],
- 1 igrišče,
- 16 sodelujočih ekip,
- 20 minut za posamezno tekmo,
- 12 ur za celotno tekmovanje.

Mali nogomet smo izbrali zaradi izkušenj organizacije tekmovanj v omenjeni športni panogi. Sistem igranja 4 + 1 je večinoma uporabljen na tekmovanjih v malem nogometu. Po navadi ima organizator na razpolago največ eno igrišče. V nasprotnem primeru lahko predpostavimo, da se tekmovanje izvede hitreje. Za 16 sodelujočih ekip smo se odločili, ker je na lokalni ravni navedena številka že nekaj ponosnega. Čas tekmovanja smo omejili na 12 ur za potrebe enodnevne prireditve. Večinoma organizatorjem ni na voljo dvorana ali nočna razsvetljava in so omejeni na dnevno svetlobo. Posamezno tekmo smo omejili na 20 minut, kar je klasična postavka na tekmovanjih v malem nogometu. Izbrani parametri so vodilo, na podlagi katerega bomo izbirali naš razporeditveni sistem.

[1] V posamezni ekipi sodelujejo golman in štirje igralci.

2.3 Svetovno prvenstvo v nogometu - primer dobre prakse

FIFA je glavna nogometna organizacija na svetu in vsaka štiri leta priredi največje nogometno tekmovanje na svetu [5]. Leta 2014 je na svetovnem prvenstvu v Braziliji sodelovalo 32 najboljših držav s celega sveta. Pred vsakim svetovnim prvenstvom FIFA organizira srečanja, na katerih določijo kriterije in principe za sodelovanje na prvenstvu in potek žrebanja. Med navedenimi principi poskrbijo za ločitev najboljših držav in geografsko lokacijo posameznih držav. Oglejmo si, kako poteka žrebanje za najprestižnejše prvenstvo na svetu.

Posamezne države razporedijo od najmočnejše do najslabše glede na dosežene točke, izračunane iz rezultatov v zadnjih štirih letih. Na podlagi izračunanih točk se v prvi boben dodajo najmočnejše države in država, v kateri poteka prvenstvo. V prvem bobnu se nahajajo nosilci skupin. Država, v kateri se odvija svetovno prvenstvo, avtomatično postane nosilka prve skupine. Ostale države se razporedijo v ostale skupine. V naslednjem bobnu se nahajajo države članice zvez CAF in CONMEBOL. V tretji boben so se dodale države članice zvez AFC in CONCACAF. V zadnjem bobnu ostanejo države članice zveze UEFA. Posamezne države se iz bobnov po vrsti žreba v skupine. V primeru, da pride do konflikta geografske lokacije, se določeno skupino preskoči. Zaradi prevelikega števila držav iz določene zveze, se v nekaterih skupinah pojavi več držav z iste geografske lokacije. Med žrebanjem skupine se določi tudi mesto v skupini, ki se uporablja za določitev terminov tekem na svetovnem prvenstvu.

Poglavje 3

Ključni problemi

Za potrebe osredotočenosti diplomske naloge smo se odločili, da bomo raziskali klasične razporeditvene sisteme. Ugotovitve raziskovanja bo možno uporabiti pri večjem številu športnih panog. Osredotočili smo se na organizacijo tekmovanja v malem nogometu in postavili določene parametre, ki si jih lahko ogledamo v tabeli 3.1 spodaj.

Tabela 3.1: Legenda parametrov

kratica	**opis**	**vrednost**
šse	število sodelujočih ekip	16 ekip
čpt	čas posamezne tekme	20 minut
čt	čas tekmovanja	12 ur * 60 minut = 720 minut
špt	število potrebnih tekem	

Zaradi želje po enodnevni prireditvi se v prvi vrsti pojavi časovna omejitev. Čas je vedno pomemben dejavnik, ki vpliva na potek tekmovanja. Na omenjeni dejavnik vpliva tudi število sodelujočih ekip. Razporeditveni sistemi imajo različno število tekem, glede na število ekip. Naslednji dejavnik je naključnost, ki zelo vpliva na potek tekmovanja in končnega zmagovalca. Potrebno je poiskati sistem, ki bo zadostoval potrebam obeh dejavnikov.

3.1 Čas

Čas je zelo pomemben dejavnik pri organizaciji enodnevne prireditve. Pri izbiri turnirskega razporeditvenega sistema moramo biti pozorni, da lahko shemo tekmovanja sploh izvedemo v okviru časovne omejitve. Izračunali bomo število tekem, potrebnih za izpeljavo klasičnih razporeditvenih sistemov za podane parametre. Na podlagi izračunov bomo izpostavili ugotovitve za posamezne sisteme.

3.1.1 Krožni sistem

Pri krožnemu sistemu igrajo sodelujoče ekipe vsaka proti vsaki. Posamezna ekipa ima število tekem za eno manjše od števila vseh sodelujočih ekip. Oglejmo si izračun na podlagi podanega algoritma:

```
1  int špt = 0;
2  int šse = 16;
3  int števec = 1;
4  for (int i = 0; i < šse; i++)
5  {
6          špt += (šse - števec);
7          števec = števec + 1;
8  }
```

$$špt = 120 \text{ tekem}$$
$$čt = 120 \text{ tekem} * 20 \text{ minut} = 2400 \text{ minut} > 720 \text{ minut}$$

Eden izmed najprimernejših sistemov, če nam je merilo pravičnosti tekmovanja, porabi ogromno časa za izpeljavo. Namesto želenih 720 minut, bi potrebovali 2400 minut, da bi izpeljali tekmovanje. Upoštevaje to dejstvo je uporaba krožnega sistema neprimerna za organiziranje enodnevne prireditve.

3.1.2 Švicarski sistem

Namesto, da bi igrale vse sodelujoče ekipe proti vsaki, imamo možnost švicarskega sistema. Ekipe se razporedijo po rangirani listi in število potrebnih tekem

se opazno zmanjša. Posamezna ekipa se sreča s polovico sodelujočih ekip. Z uporabo naslednjega algoritma lahko izračunamo število potrebnih tekem za izpeljavo omenjenega sistema:

```
int špt = 0;
int še = 16;
int števec = še;
while (števec > 1)
{
        špt += (še / 2);
        števec = števec / 2;
}
```

$$špt = 32 \text{ tekem}$$
$$čt = 32 \text{ tekem} * 20 \text{ minut} = 640 \text{ minut} < 720 \text{ minut}$$

Izračun števila tekem nam je pokazal, da je sistem primeren glede na časovno omejitev.

3.1.3 Izločilni sistem

Sistem izločanja je najhitrejši sistem za izpeljavo tekmovanja. Posamezna ekipa potrebuje samo zmage za napredovanje. Zaradi tega je število tekem na posamezno ekipo raznoliko. Problem omenjenega sistema se izpostavi v pravičnosti do sodelujočih ekip na tekmovanju. Z naslednjim algoritmom lahko izračunamo število potrebnih tekem:

```
int špt = 0;
int še = 16;
while (še > 1)
{
        še = še / 2;
        špt += še;
}
```

$$špt = 15 \text{ tekem}$$
$$čt = 15 \text{ tekem} * 20 \text{ minut} = 300 \text{ minut} < 720 \text{ minut}$$

Zaradi izločanja ekip se število potrebnih tekem opazno zmanjša in glede časovne omejitve je sistem primeren za uporabo. V nadaljevanju si bomo ogledali, kako naključnost vpliva na organiziranje tekmovanja.

3.2 Naključnost

Težnja vsakega organizatorja je izbira pravičnega zmagovalca na posameznemu tekmovanju. Razporeditveni sistemi med žrebom samim vključujejo dejavnik naključnosti. Potrebno je poiskati primerno stopnjo naključnosti za izvedbo zanimivega tekmovanja.

3.2.1 Krožni sistem

Dejavnik naključnosti pri krožnemu sistemu bistveno izgubi pomen, ker se sodelujoče ekipe pomerijo vsaka proti vsaki. Edina naključnost temelji na razporedu časovnega termina posameznih tekem.

3.2.2 Švicarski sistem

Pri švicarskem sistemu se zaradi rangirane liste večinoma ne poslužujemo naključja. Ekipe so razporejene na listi od najboljše do najslabše glede na število doseženih točk. Na naključnost vplivajo le nove ekipe, ki se po navadi dodajo na dno lestvice.

3.2.3 Izločilni sistem

Izločilni sistem temelji na naključju. Pred začetkom tekmovanja se opravi žreb, ki določi shemo, po kateri se odigrajo medsebojni dvoboji. Poleg posameznih dvojic vpliva žreb tudi na časovni razpored tekem. Zaradi naključnosti lahko pride med žrebom do problemov skupine smrti in lokacije ekip.

3.3 Skupina smrti

Glavni problem naključnosti žreba je t.i. skupina smrti. Med žrebom se lahko hitro dogodi, da se v določeni skupini srečajo najboljše ekipe na tekmovanju. V navedenem primeru se zanimivost tekmovanja zmanjša in naključnost premočno vpliva na zmagovalca. Zaradi vnaprej določene sheme tekmovanja, se omenjenega problema izogneta krožni in švicarski sistem. V prikazano past se ujame izločilni sistem, ki temelji na naključnosti žreba. Lahko se zgodi, da se že v prvem krogu izločijo najboljše ekipe.

3.4 Lokacija ekip

Na tekmovanjih sodelujejo ekipe iz različnih krajev. Med žrebom se lahko zgodi, da se že na začetku srečajo ekipe iz istih lokacij. Za zanimivost poskrbimo, če se omenjene ekipe lahko srečajo proti koncu tekmovanja. V skupinah, kjer se nahajajo ekipe iz istih lokacij, lahko pride do špekulacij pri rezultatih.

3.4.1 Krožni sistem

Glede na to, da igrajo ekipe vsaka proti vsaki, lokacija vpliva proti koncu tekmovanja. Ekipe iz istih lokacij lahko na zadnjih tekmah lažje prirejajo rezultate za boljšo uvrstitev določene ekipe. V krožnemu sistemu je zaradi tega primerno, da ekipe iz istih lokacij odigrajo tekme že v prvi polovici tekmovanja.

3.4.2 Švicarski sistem

Razpored tekem je vnaprej določen in ne moramo vplivati, če se sodelujoče ekipe iz istih lokacij srečajo že na začetku tekmovanja.

3.4.3 Izločilni sistem

Sistem je naključen in od žreba je odvisno, kdaj se srečajo ekipe iz istih lokacij. Obstaja možnost, da se omenjene ekipe srečajo že na začetku.

3.4.4 Ugotovitve

Med raziskovanjem klasičnih razporeditvenih sistemov smo ugotovili, da moramo paziti na določene pasti. Ugotovili smo, da je najpravičnejši krožni sistem. S tekmovanjem najhitreje opravimo, če uporabimo izločilni sistem. Problemov glede skupine smrti se rešimo z uporabo švicarskega sistema. Za lokacijo ekip lahko poskrbimo, če preskakujemo skupine z ekipami iz istih lokacij. Navedene ugotovitve so nas pripeljale do spoznanja, da potrebujemo kombinacijo klasičnih sistemov. Odločili smo se uporabiti sistem, ki se tradicionalno uporablja na svetovnem prvenstvu, in ga prirediti našim potrebam.

Odločili smo se tekmovanje razdeliti na skupinski in izločilni del. Med skupinskim delom ekipe odigrajo tekme vsaka proti vsaki. Iz posamezne skupine napredujeta dve najboljši ekipi. V izločilnem delu se pomerijo ekipe v medsebojnih dvobojih, dokler ne ostane na vrhu zmagovalna ekipa. Za določanje skupin se bomo poslužili rangirne liste, s katero bomo reševali probleme skupine smrti. Z navedeno listo bomo določili nosilce skupin. Preostale ekipe se nato do zadnje žrebajo v skupine, da poskrbimo za določeno raven naključnosti na tekmovanju. V naslednjem poglavju si oglejmo, kako bomo nastavili rangirno listo in izpeljali žreb tekmovanja.

3.5 Načrtovanje tekmovanja z uporabo sistema ELO

Arpad Elo je bil izumitelj sistema ELO [6], ki se je prvotno uporabljal v šahu. Sedaj se omenjeni sistem uporablja tudi v drugih športih, igrah in celo tekmovalnemu programiranju. Sistem se uporablja tudi na svetovnemu prvenstvu, kar pomeni določeno uporabnost v našem primeru. Izpostaviti

moramo dejstvo, da se na svetovnemu prvenstvu točkuje ekipa. Pri našem primeru lahko ekipe med tekmovanji zamenjajo ime ali igralce. Za navedeno problematiko obstaja rešitev, katere se poslužujejo internetne igre (kot je `League of Legends`). Internete igre predpostavljajo, da igralci ne bodo vedno v istih ekipah in vodijo točke ELO za igralca, namesto za celotne ekipe. Med ustvarjanjem nove igre, so igralci določeni v ekipo in točke ekipe postanejo enake povprečnim točkam igralcev.

V našem primeru morajo ekipe prijaviti igralce, za katere izračunamo povprečje točk. Ekipe lahko nato razporedimo od najboljše do najslabše. Za igralce brez točk določimo začetno število točk. Po ustvarjeni rangirni listi lahko določimo nosilce skupin. Število nosilcev je enako številu skupin. V skupine najprej žrebamo nosilce, zatem žrebamo v skupine še preostale ekipe. Seveda lahko med žrebom pride do problemov zaradi lokacij ekip. Ta problem rešimo s tem, da vedno prestavimo ekipo v naslednjo skupino, ki ima najmanjše število ekip iz iste lokacije. Med žrebanjem posamezne ekipe lahko še žrebamo pozicijo v skupini in s tem poskrbimo za naključni vrstni red tekem. Shema tekmovanja je še vedno odvisna od organizatorja tekmovanja samega.

3.5.1 Primer žreba

Tabela 3.2: Lestvica sodelujočih ekip

ekipa	**lokacija**	**točke**	**ekipa**	**lokacija**	**točke**
ekipa1	lokacija1	116	ekipa9	lokacija9	108
ekipa2	lokacija2	115	ekipa10	lokacija10	107
ekipa3	lokacija3	114	ekipa11	lokacija11	106
ekipa4	lokacija4	113	ekipa12	lokacija12	105
ekipa5	lokacija5	112	ekipa13	lokacija13	104
ekipa6	lokacija6	111	ekipa14	lokacija14	103
ekipa7	lokacija7	110	ekipa15	lokacija1	102
ekipa8	lokacija8	109	ekipa16	lokacija1	101

Na podani hipotetični lestvici sodelujočih ekip v tabeli 3.2, si oglejmo primer izpeljave celotnega žreba. Pri šestnajstih sodelujočih ekipah lahko ustvarimo štiri skupine, v katerih bodo po štiri ekipe. Najprej določimo štiri nosilce, ki so ekipe z največ točkami in jih naključno razporedimo v različne skupine. S tem poskrbimo, da se na začetku tekmovanja ne morejo srečati najboljše ekipe. Oglejmo si primer reševanja problema skupine smrti v tabeli 3.3:

Tabela 3.3: Reševanje problema skupine smrti

skupina 1	**skupina 2**	**skupina 3**	**skupina 4**
ekipa3	ekipa2	ekipa1	ekipa4

Nato razporedimo še preostale ekipe. V primeru, da se soočimo s problemom lokacije, preskočimo skupino in nadaljujemo z žrebom. V tabeli 3.4 si lahko ogledamo opravljen preskok zaradi lokacije petnajste ekipe:

Tabela 3.4: Reševanje problema lokacije

skupina 1	**skupina 2**	**skupina 3**	**skupina 4**
ekipa3	ekipa2	ekipa1	ekipa4
ekipa16	ekipa5	ekipa14	ekipa10
ekipa9	ekipa8		ekipa15

V tabeli 3.5 si oglejmo zaključen žreb:

Tabela 3.5: Zaključen žreb

skupina 1	**skupina 2**	**skupina 3**	**skupina 4**
ekipa3	ekipa2	ekipa1	ekipa4
ekipa16	ekipa5	ekipa14	ekipa10
ekipa9	ekipa8	ekipa7	ekipa15
ekipa11	ekipa13	ekipa6	ekipa12

Poglavje 4

Razvoj mobilne aplikacije

Za potrebe preverjanja omenjenih sistemov smo se odločili razviti mobilno aplikacijo, preko katere bomo prikazali dodatne ugotovitve. Med razvijanjem aplikacije smo se še globlje spoznali z celotno tematiko in lažje razumeli potrebe organizatorjev. Aplikacijo smo poimenovali `Tournament Coordinator` za voljo širše kulturne prepoznavnosti.

Preden smo začeli pisati izvorno kodo, smo proučili zahteve naše aplikacije in opredelili njeno uporabo. Poleg tega smo morali proučiti, kakšen podatkovni model potrebujemo in določiti našo strukturo. Po analizi smo začeli z implementacijo izvorne kode.

4.1 Uporabljene tehnologije in orodja

Za preverjanje posameznih sistemov, smo razvili aplikacijo. Za mobilno aplikacijo smo se odločili zaradi večje uporabnosti pri končnih uporabnikih. Za platformo smo uporabili Android [7] zaradi njegove razširjenosti in prepoznavnosti. Orodje, ki nam je omogočilo izdelavo aplikacije, je bil Android Studio verzije 1.3.2 [8]. Med delom smo vodili zgodovino verzij z vtičnikom Git [9], ki nam je omogočil varen razvoj in hranjenje izvorne kode. Za potrebe hranjenja podatkov smo uporabili SQLite bazo [10].

4.1.1 Platforma Android

Android je operacijski sistem za pametne telefone, ter ostale prenosne naprave [7]. Zgrajen je na Linuxovem jedru. Android je odprtokoden sistem, zaradi česar omogoča lažji razvoj programske opreme. Sistem, podoben Unix-u je enostaven, odziven in omogoča večopravilnost. Omogoča zagon aplikacij v ozadju, ki lahko uporabniku med delom še vedno pošiljajo informacije. Operacijski sistem Android je sestavljen iz petih elementov:

- aplikacije,
- aplikacijsko ogrodje,
- knjižnice,
- prevajalnik,
- Linux kernel.

Aplikacije so napisane v programskem jeziku Java. Posamezna aplikacija večinoma vsebuje različne aktivnosti. Na aktivnosti se določi njena funkcionalnost. Delovanje aplikacije poteka večinoma preko menjave aktivnosti v ozadje in ospredje. Določiti je potrebno hierarhijo, kjer si aktivnosti sledijo v primernem vrstnem redu.

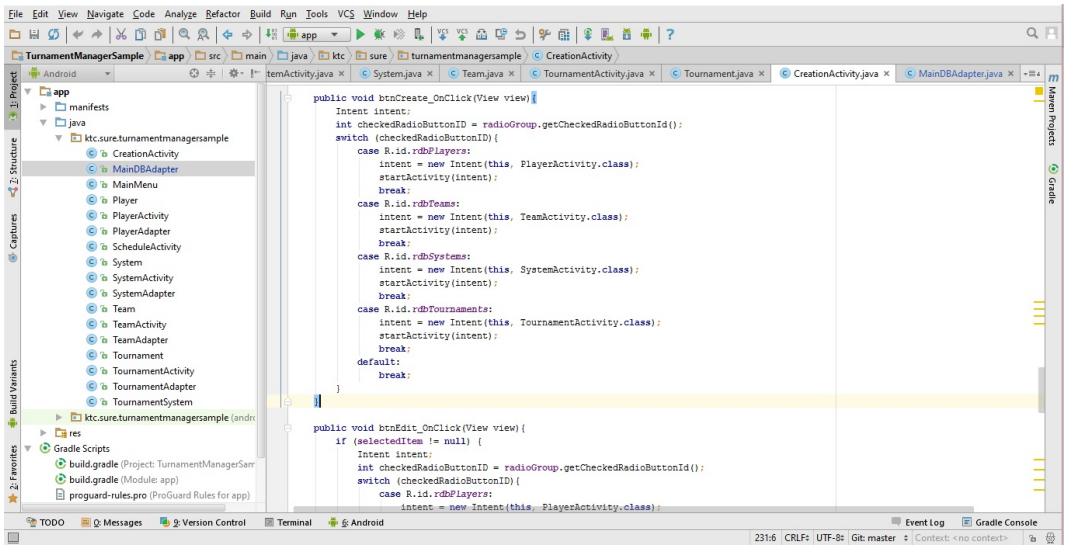
4.1.2 Programska knjižnica SQLite

SQLite je najboljši približek podatkovne baze za Android sistem [10]. Vgrajena funkcionalnost poskrbi za hitro in odzivno delo s podatki. Za delo uporabi tekstovno datoteko in preko nje uporablja celotno knjižnico, ki vsebuje ogromno metod, ki poskrbijo za delo s tekstovno datoteko. Omogoča zelo majhno porabo spomina, ki se poveča, če naprava to omogoča in s tem izboljša funkcionalnost. Deluje na različnih platformah, kar omogoča lažjo prenosljivost in primerno izbiro za aplikacije. `SQLite` poskrbi za zanesljivost in preprosto uporabo. Njihovi razvijalci so zavoljo enostavnosti izpustili

določene funkcionalnosti in omogočili uporabniku, da z nekaj vrsticami kode že postavi svojo podatkovno strukturo.

4.1.3 Vgrajeno razvijalno okolje Android Studio 1.3.2

Android Studio je glavno orodje za delo z Android operacijskim sistemom in ponuja vse, kar lahko potrebujemo za izdelavo aplikacije za pametni telefon [8]. Orodje je intuitivno za uporabo in omogoča uporabniku preprosto načrtovanje aplikacije z dodajanjem komponent preko izbirnega menija. Med ustvarjanjem novega projekta imamo možnost izbire med aktivnostmi, ki nam prinašajo različne funkcionalnosti. Menuje si lahko razporedimo po lastni potrebi in si omogočimo delovanje po našem okusu. Na sliki 4.1 si lahko ogledamo primer odprtega projekta v `Android Studiu`. Preprosto lahko preklopimo med urejevalnikom vmesnika in izvorne kode.



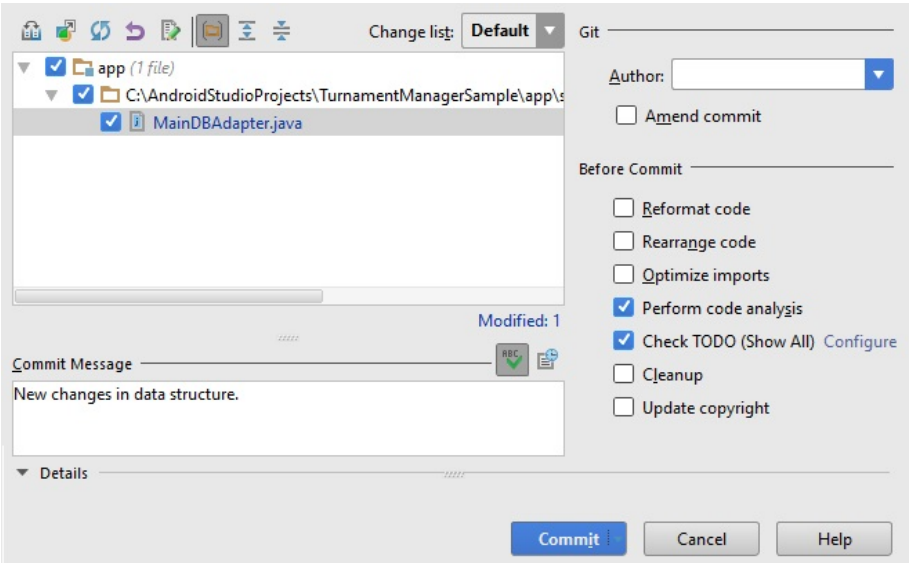
Slika 4.1: Primer projekta v `Android Studiu`.

4.1.4 Repozitorij izvorne kode Git

Git je brezplačni odprtokodni sistem za shranjevanje različic programske kode malih ali velikih projektov [9]. Navedeni sistem poskrbi, da se vedno lahko vrnemo nazaj na verzijo, ki jo želimo uporabiti. Predstavljamo si, da smo med pisanjem programske kode malo zašli in po nepotrebnem pokvarili našo strukturo. Brez hranjenja različic je lahko prava nočna mora povrniti delujoče stanje. S sistemom, kot je `Git` se lahko brez problema vrnemo nazaj na delujočo verzijo. Večina podobnih sistemov se poslužuje shranjevanja vseh datotek. `Git` naredi le posnetek do že shranjenih datotek za hitrejšo uporabo. Na sliki 4.2 si lahko ogledamo primer posodobitve datoteke.

Postopek delovanja:

- izvorna koda se spremeni.
- spremembe se dodajo v čakalno listo.
- opravi se posodobitev v direktoriju `Git`.



Slika 4.2: Posodobitev datoteke v repozitoriju `Git`.

4.2 Analiza funkcionalnosti

Analiza splošne problematike nam je pokazala, da je osnovna uporabniška zahteva ustvarjanje in urejanje novih turnirjev. Po tvorbi novega turnirja mora biti uporabniku omogočena možnost pregleda opravljenega žreba in ponovitev žreba. Za boljšo uporabniško izkušnjo je potrebno pokriti čimvečje število športnih panog. Zatorej smo se odločili, da bodo uporabniki imeli možnost ustvarjanja svojih sistemov glede na podane nastavitve. Zavoljo implementacije ekipnih športov, je bilo potrebno omogočiti urejanje igralcev in ekip ter dodeljevanje teh posameznim turnirjem. Primerno je, da omogočimo dodeljevanje igralcev posameznim ekipam.

Na podlagi analize splošne uporabnosti smo se odločili realizirati naslednje funkcionalne zahteve:

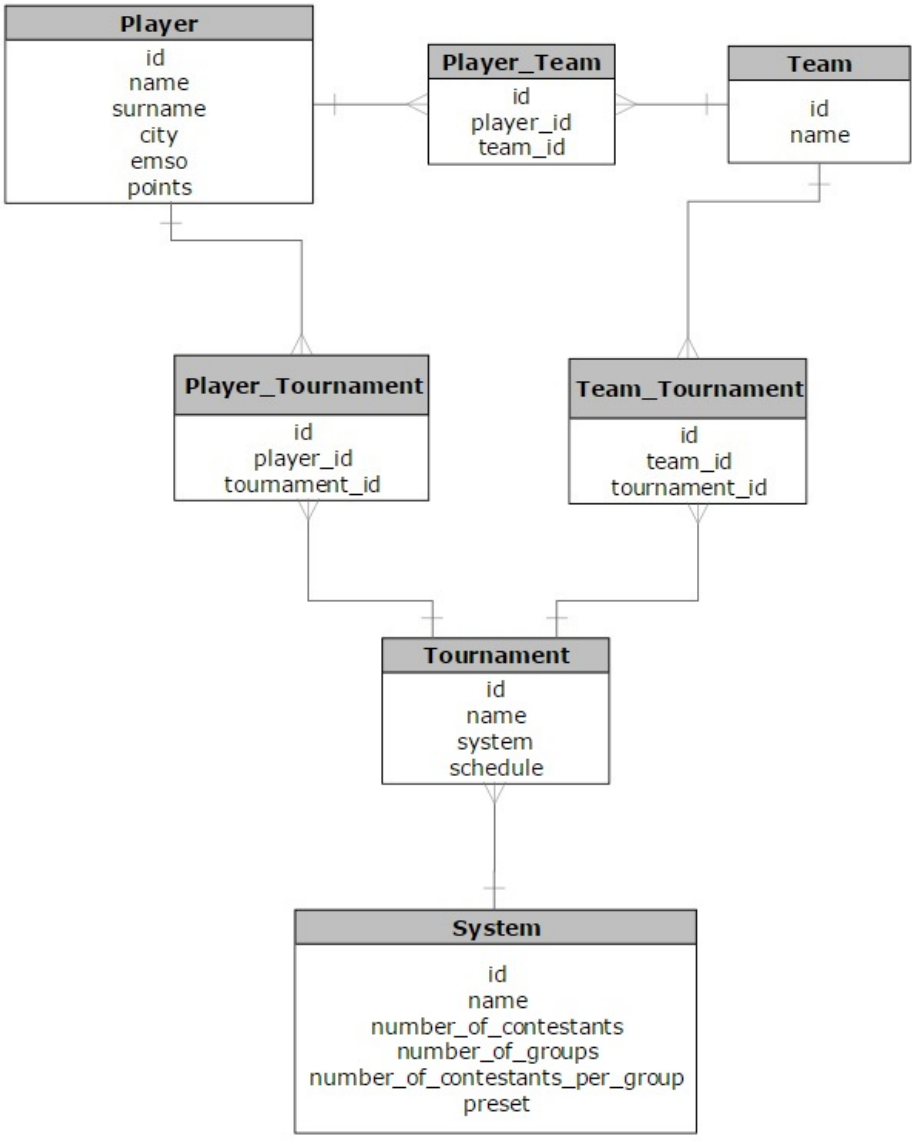
- ustvarjanje novih igralcev, ekip, sistemov, tekmovanj in žrebov
- dodeljevanje igralcev posamezni ekipi,
- dodeljevanje sistemov in igralcev ali ekip k posameznim tekmovanjem,
- pregled igralcev, ekip, sistemov in tekmovanj,
- urejevanje igralcev, ekip, sistemov in tekmovanj,
- pregled žrebov.

Poleg funkcionalnih smo se odločili realizirati še naslednje nefunkcionalne zahteve:

- intuitiven in uporabniku prijazen uporabniški vmesnik,
- uvoz podatkov iz `.xls` datoteke,
- izvoz podatkov v `.xls` datoteko,
- izvoz pregleda žreba v `.txt` datoteko.
- za varnost podatkov je poskrbljeno preko dejstva, da je aplikacija dostopna uporabniku pametnega telefona.

4.3 Načrtovanje podatkovnega modela

Na podlagi analize zahtev smo ugotovili, da potrebujemo podatkovni model, sestavljen iz sedmih medsebojno povezanih tabel. Na sliki 4.3 si lahko ogledamo shemo relacijskega podatkovnega modela:



Slika 4.3: Relacijski podatkovni model.

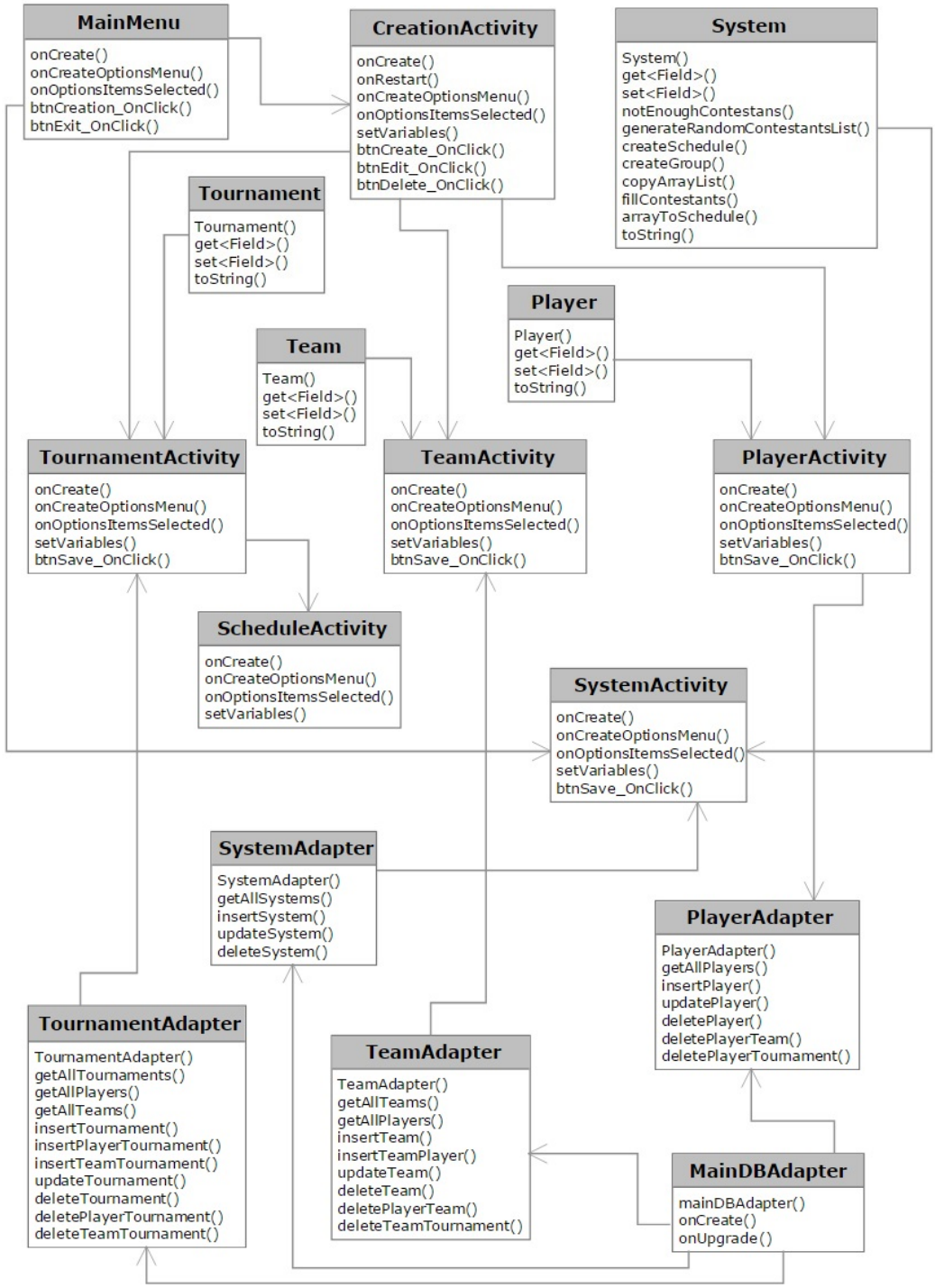
Podatki o igralcih, ekipah in tekmovanjih se bodo shranjevali v tabele `Player`, `Team` in `Tournament`, ki so v relaciji mnogo proti mnogo. Za povezovalne tabele smo se odločili uporabiti `Player_Team`, `Player_Tournament` in `Team_Tournament`. Posamezen igralec lahko igra za več ekip. V posamezni ekipi je ponavadi več, kot en igralec. Tekmovanja se lahko udeleži več igralcev ali ekip. Posamezen igralec ali ekipa lahko sodeluje na mnogih tekmovanjih. V tabelo `System` se bodo shranjevali razporeditveni sistemi, ki jih določi uporabnik sam. S tem nudimo možnost uporabe v različnih športnih panogah. Na tekmovanju se lahko uporabi en razporeditveni sistem, vendar se lahko sistem uporabi na več tekmovanjih.

4.4 Opis rešitve

Po pregledu zahtev, analizi uporabe in načrtovanju podatkovnega modela smo začeli z implementacijo rešitve. Ogledali si bomo glavne sestavine mobilne aplikacije s poudarkom na tem, kako smo rešili posamezne probleme.

4.4.1 Sestava razredov

Poskrbeti smo želeli za ustrezno sestavo funkcionalnosti, ki se odražajo v hierarhični sestavi razredov. Na najvišjem nivoju se nahaja `MainMenu`, ki skrbi za navigacijo po posameznih skupinah uporabe. Prvi sklop uporabe smo zapakirali v `CreationAcitivity`, kjer lahko uporabnik ustvarja, ureja in briše posamezne vnose. Glavni del podatkovne baze smo zapakirali v razdred `MainDBAdapter`, ki skrbi za tvorbo in brisanje tabel. Določili smo glavne skupine: `player`, `team`, `tournament` in `system`. Glavnim skupinam smo dodelili tri razrede, ki skrbijo za tekoče delovanje in komuniciranje med posameznimi skupinami. Na sliki 4.4 si lahko ogledamo osrednjo sestavo razredov:



Slika 4.4: Razredni digram.

4.5 Podroben prikaz vsebine aplikacije

Na najvišjem nivoju smo posamezne sklope funkcionalnosti razdelili na naslednje faze:

- tvorba,
- prikaz,
- urejanje,
- brisanje.

V nadaljevanju si bomo ogledali tipične primere rešitve.

4.5.1 Tvorba

Za ustvarjanje igralcev, ekip, sistemov in turnirjev se pokliče metoda `btnCreate_OnClick`.

```
1  public void btnCreate_OnClick(View view){...}
```

Glede na uporabnikovo izbiro skupine, se zažene primerna vsebina.

```
1  switch (checkedRadioButtonID){
2          case R.id.rdbPlayers:
3                  intent = new Intent(this, PlayerActivity.class);
4                  startActivity(intent);
5                  break;
6          case R.id.rdbTeams:
7                  intent = new Intent(this, TeamActivity.class);
8                  startActivity(intent);
9                  break;
10         case R.id.rdbSystems:
11                 intent = new Intent(this, SystemActivity.class);
12                 startActivity(intent);
13                 break;
14         case R.id.rdbTournaments:
15                 intent = new Intent(this, TournamentActivity.class);
16                 startActivity(intent);
17                 break;
18         default:
19                 break;
20 }
```

Vsaka skupina ima svoje metode za tvorbo novega elementa. Med shranjevanjem v tabelo se pokliče metoda `btnSave_OnClick`.

```
public void btnSave_OnClick(View view) {...}
```

Na primeru igralca si oglejmo tvorbo novega elementa. Najprej se ustvari nov objekt tipa `player`.

```
currentPlayer = new Player();
```

Nato se preveri, če je uporabnik izpolnil vsa potrebna polja.

```
if (name.isEmpty() || name == ""
|| surname.isEmpty() || surname == ""
|| city.isEmpty() || city == ""
|| emso.isEmpty() || emso == ""
|| points <= -1
) {
  textView.setText("Some fields are missing");
  return;
}
```

Novemu igralcu se dodelijo ustrezne vrednosti.

```
currentPlayer.setName(name);
currentPlayer.setSurname(surname);
currentPlayer.setCity(city);
currentPlayer.setEmso(emso);
currentPlayer.setPoints(points);
```

Nato se pokliče metoda za urejanje igralca.

```
long id = playerAdapter.updatePlayer(currentPlayer);
```

V metodi `updatePlayer` se preveri, če gre za novega igralca. V tem primeru se pokliče metoda za tvorbo igralca `insertPlayer`.

```
if (player.getId() == -1) {
  return insertPlayer(player);
}
```

Metoda `insertPlayer` napolni vnešene podatke o igralcu in jih zapiše v bazo. Nazaj nam vrne ključ novega igralca ob uspešnem vnosu ali vrednost -1 ob neuspešnem vnosu.

```
public long insertPlayer(Player player)
{
  SQLiteDatabase db = this.getWritableDatabase();
  ContentValues contentValues = new ContentValues();
  contentValues.put("name", player.getName());
  contentValues.put("surname", player.getSurname());
  contentValues.put("city", player.getCity());
  contentValues.put("emso", player.getEmso());
  contentValues.put("points", player.getPoints());
  return db.insert("player", null, contentValues);
}
```

Po vnosu v bazo metoda `btnSave_OnClick` preveri, če je zapis dejansko uspel in o tem obvesti uporabnika.

```
if (id > -1) {
  textView.setText("Player " + currentPlayer + " was saved");
  setVariables(false, true);
} else {
  textView.setText("Player " + currentPlayer + " was not saved");
}
```

V primeru uspešnega vnosa se pokliče metoda `setVariables`, ki počisti vnosna polja in tvori nov objekt tipa `player`.

```
if (clear) {
  edtName.setText("");
  edtSurname.setText("");
  edtCity.setText("");
  edtEmso.setText("");
  edtPoints.setText("");

  currentPlayer = new Player();
}
```

Za ekipo, sistem in tekmovanje poteka tvorba novega elementa na podoben način, z uporabo dodatnih parametrov.

4.5.2 Prikaz

Na nivoju `CreationActivity` se uporabniku glede na izbrano skupino prikažejo spremembe v podatkovni bazi.

```
int checkedRadioButtonID = radioGroup.getCheckedRadioButtonId();
switch (checkedRadioButtonID){
  case R.id.rdbPlayers:
    listView.setAdapter(aaPlayers);
    break;
  case R.id.rdbTeams:
    listView.setAdapter(aaTeams);
    break;
  case R.id.rdbSystems:
    listView.setAdapter(aaSystems);
    break;
  case R.id.rdbTournaments:
    listView.setAdapter(aaTournaments);
    break;
  default:
    listView.setAdapter(aaPlayers);
    break;
}
```

Za prikaz seznama igralcev se pokliče metoda `getAllPlayers`, ki iz podatkovne baze vrne objekt tipa `ArrayList` z vsemi vnosi v tabeli `player`.

```
if (load_players) {
  alPlayers = playerAdapter.getAllPlayers();
  aaPlayers =
    new ArrayAdapter(this,
                     android.R.layout.simple_spinner_item,
                     alPlayers);
}
```

V metodi `getAllPlayers` se ustvari nov tip objekta `ArrayList`.

```
ArrayList<Player> arrayList = new ArrayList();
```

Nato se izvede poizvedba, s katero pridobimo vse vnose iz tabele `player`.

```
SQLiteDatabase db = this.getReadableDatabase();
Cursor res =  db.rawQuery( "select * from player", null );
res.moveToFirst();
```

Seznam igralcev se napolni in vrne z vsemi pridobljenimi elementi.

```
Player currentPlayer;

while(res.isAfterLast() == false){
  currentPlayer = new Player(id,
                             name,
                             surname,
                             false,
                             city,
                             emso,
                             points);
  arrayList.add(currentPlayer);
  res.moveToNext();
}

res.close();
return arrayList;
```

4.5.3 Urejanje

Urejanje igralcev, ekip, sistemov in turnirjev se omogoči v metodi `btnEdit_OnClick`, ki izvrši ustrezno metodo s parametri, prebranimi iz podatkovne baze.

```
public void btnEdit_OnClick(View view){...
  if (selectedItem != null) {
    switch (checkedRadioButtonID){
      case R.id.rdbPlayers:
        intent = new Intent(this, PlayerActivity.class);
        intent.putExtra("Player", (Player) selectedItem);
        startActivity(intent);
        break;
      case R.id.rdbTeams:
        intent = new Intent(this, TeamActivity.class);
        intent.putExtra("Team", (Team) selectedItem);
        startActivity(intent);
        break;
      ...
      default:
        break;
    }
  }
}
```

Za izbranega igralca se usvari nov objekt tipa `player`, v katerega se naložijo podatki iz podatkovne baze.

```
currentPlayer = (Player) getIntent().getSerializableExtra("Player");
}
```

Vnosna polja se napolnijo s podatki izbranega igralca.

```
edtName.setText(currentPlayer.getName());
edtSurname.setText(currentPlayer.getSurname());
edtCity.setText(currentPlayer.getCity());
edtEmso.setText(currentPlayer.getEmso());
```

Nato se opravi preverjanje podatkov, kot pri ustvarjnaju novega igralca. Pokliče se metoda `updatePlayer`, ki poskrbi za posodobitev podatkov za izbranega igralca v tabeli `player`.

```
SQLiteDatabase db = this.getWritableDatabase();
ContentValues contentValues = new ContentValues();
contentValues.put("name", player.getName());
contentValues.put("surname", player.getSurname());
contentValues.put("city", player.getCity());
contentValues.put("emso", player.getEmso());
contentValues.put("points", player.getPoints());
return db.update("player", contentValues, "id = ? ", new
 ↪  String[]{Integer.toString(player.getId())});
```

Po vnosu se opravi preverjanje, če so bili novi podatki vnešeni uspešno.

4.5.4 Brisanje

Za brisanje izbranega elementa poskrbi metoda `btnDelete_OnClick`. Izbrani element se odstrani iz podatkovne baze. Seznam elementov se uporabniku prikaže posodoboljen.

```
public void btnDelete_OnClick(View view){...
  if (selectedItem != null) {
    int removedRows;
    switch (checkedRadioButtonID){
      case R.id.rdbPlayers:
```

```
6           removedRows =
7             playerAdapter.deletePlayer((Player) selectedItem);
8           if (removedRows > 0) {
9             aaPlayers.remove(selectedItem);
10            setVariables(false, false, false, false, false, true);
11          }
12          break;
13        case R.id.rdbTeams:
14          removedRows =
15            teamAdapter.deleteTeam((Team) selectedItem);
16          if (removedRows > 0) {
17            aaTeams.remove(selectedItem);
18            setVariables(false, false, false, false, false, true);
19          }
20          break;
21          ...
22          default:
23            break;
24      }
25    }
26  }
```

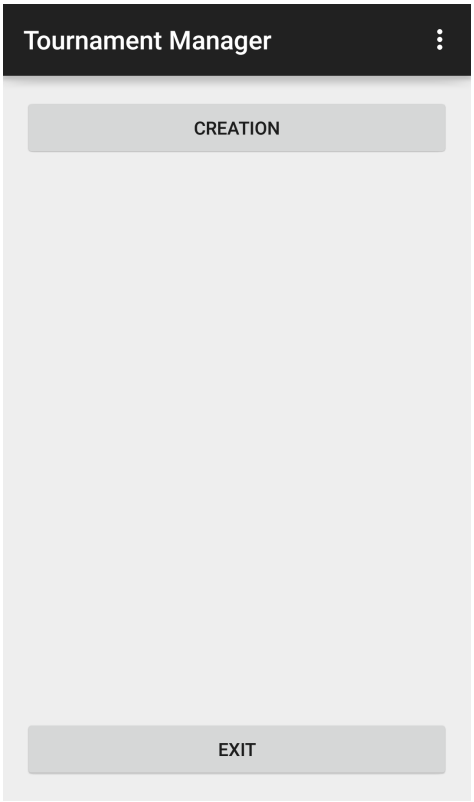
Metoda `deletePlayer` poskrbi za odstranitev igralca in vseh povezav do posameznih ekip in tekmovanj iz podatkovne baze.

```
1  SQLiteDatabase db = this.getWritableDatabase();
2  deletePlayerTeam(player);
3  deletePlayerTournament(player);
4  return db.delete("player",
5                   "id = ? ",
6                   new String[]{Integer.toString(player.getId())});
```

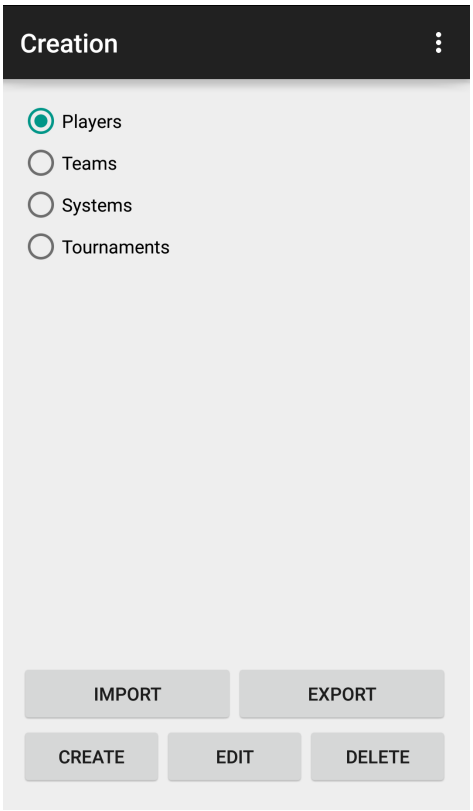
4.6 Opis uporabniškega vmesnika

Uporabniku se najprej prikaže izbor funkcionalnosti aplikacije. Zaenkrat je to le opcija `Creation`, ki vsebuje vse potrebno, da organizator uspešno izvede žreb na tekmovanju. Akcija klika na gumb `CREATION` popelje uporabnika na naslednji nivo, ki odpira nove možnosti uporabe. Na tem nivoju lahko uporabnik ustvarja, ureja in briše igralce, ekipe, sisteme ter tekmovanja. Z izbiro s seznama skupin uporabnik izbere, kaj hoče urejati. S pritiskom na

gumb `CREATE` se uporabniku odpre nova aktivnost za dodajanje novega vnosa v željeno skupino.



Slika 4.5: `MainMenu`.

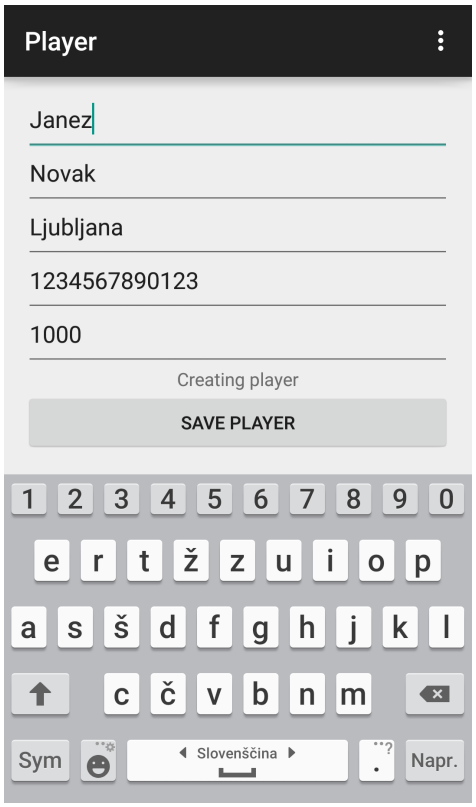


Slika 4.6: `CreationActivity`.

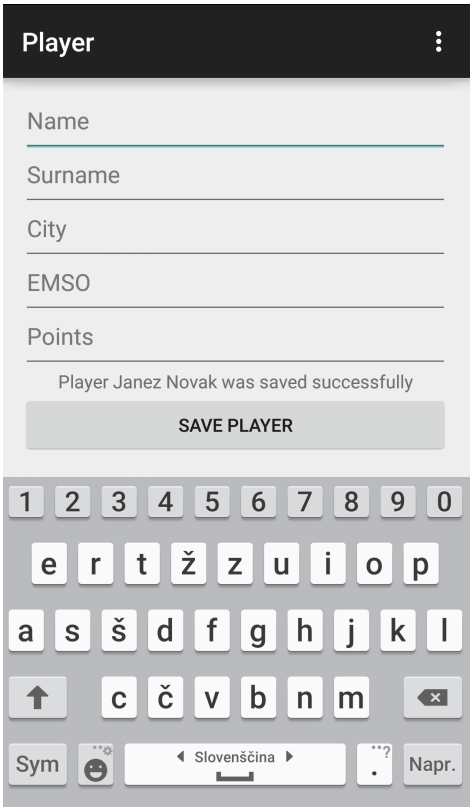
4.6.1 Ustvarjanje novega igralca

Uporabnik, ki želi ustvariti novega igralca, akcijo zažene s pritiskom na gumb `CREATE` in v primeru pravilne izbire s seznama skupin odpre aktivnost, ki omogoča vnos podatkov za nov zapis. Na aktivnosti se nahajajo tekstovna polja, ki jih uporabnik mora izpolniti. Uporabnika opozorimo, če vseh polj ne izpolni pravilno. Polja, predvidena kot števila, smo omejili na znake, ki vsebujejo le številke. Po prikazu usmerimo pozornost na prvo tekstovno polje za vnos imena in prikažemo tipkovnico. Uporabnik lahko po izpolnjevanju sproži akcijo shranjevanja novega zapisa igralca s pritiskom na gumb `SAVE`

PLAYER. Uporabniku sporočimo, če se je nov zapis uspešno shranil v podatkovno strukturo. V nasprotnem primeru ga opozorimo, da je prišlo med shranjevanjem do napake.



Slika 4.7: Ustvarjanje igralca.

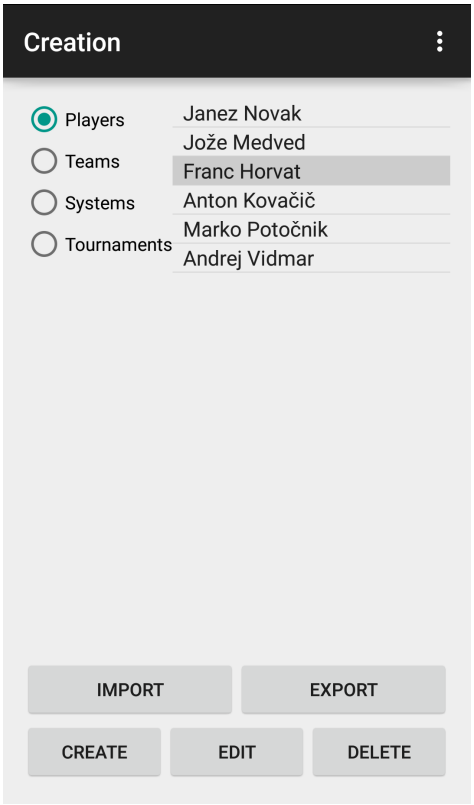


Slika 4.8: Shranjevanje igralca.

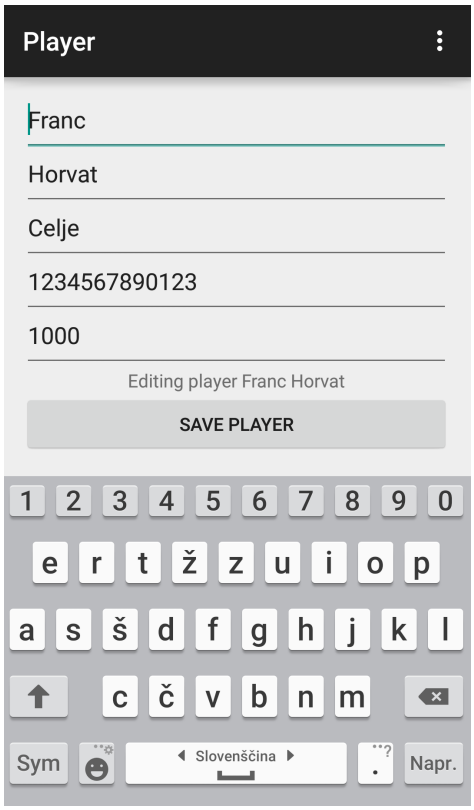
4.6.2 Urejanje obstoječih elementov

Uporabniku se na nivoju Creation v seznamu, ki se nahaja na desni strani, prikažejo vsi ustvarjeni elementi izbrane skupine. S pritiskom na posamezen element v seznamu omogočimo uporabniku možnost urejanja ali brisanja. S pritiskom na gumb DELETE se v podatkovni strukturi izbriše vnos izbranega elementa in posodobi prikazan seznam. S pritiskom na gumb EDIT se zažene nova aktivnost, ki omogoča uporabniku urejanje že obstoječega vnosa v podatkovni strukturi. Uporabnik lahko spreminja podatke in jih nato s klikom

na gumb `SAVE PLAYER` v podatkovni strukturi posodobi. Po vrnitvi na višji nivo lahko opazi, da se posodobi tudi seznam, v katerem se nahaja shranjeni element. Na seznamu elementov lahko uporabnik opravi uvoz ali izvoz podatkov s pritiskom na gumba `IMPORT` in `EXPORT`. Za to nalogo se uporabljajo skupinam pripadajoče datoteke tipa .xls. Uvoz in izvoz podatkov je uporaben za varnostno hranjevanje in arhiviranje podatkov.



Slika 4.9: Seznam igralcev.

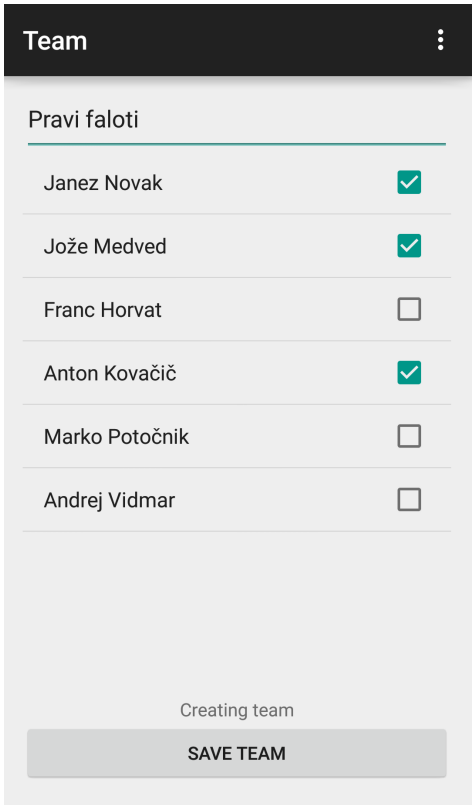


Slika 4.10: Urejanje igralca.

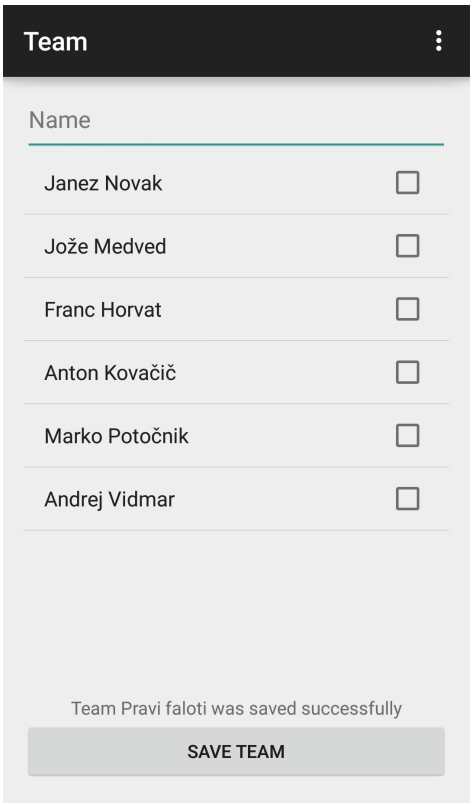
4.6.3 Ustvarjanje nove ekipe

Uporabnik lahko ustvari novo ekipo s pravilno izbiro s seznama skupin in klikom na gumb `CREATE`. Prikaže se aktivnost, v kateri se lahko vnese ime nove ekipe in dodeli igralce. Uporabniku je vedno omogočena možnost spreminjanja imena ekipe ali dodajanje in odstranjevanje izbranih igralcev. Izbrisani

igralec je avtomatično odstranjen iz ekip, katerimi je bil dodeljen. Igralce s seznama s klikom dodamo in odstranimo. S pritiskom na gumb `SAVE TEAM` shranimo nov zapis za ekipo v podatkovno strukturo in dodelimo izbrane igralce novo ustvarjeni ekipi. Uporabnika obvestimo o uspehu shranjevanja nove ekipe in opozorimo, če je prišlo do napake pri shranjevanju.



Slika 4.11: Ustvarjanje ekipe.

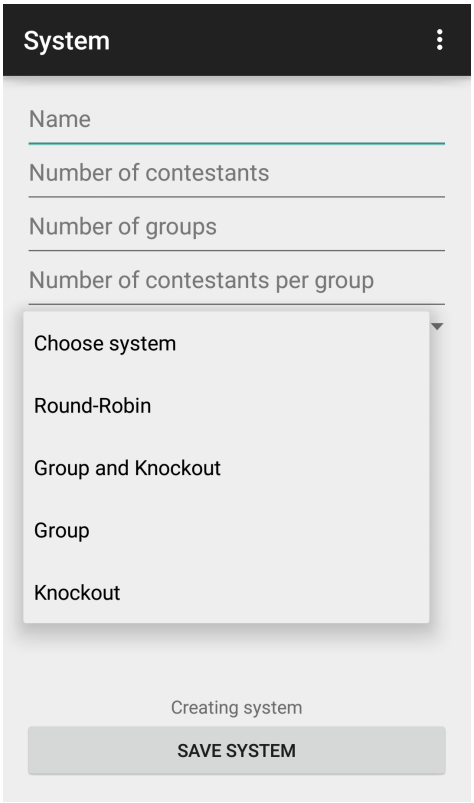


Slika 4.12: Shranjevanje ekipe.

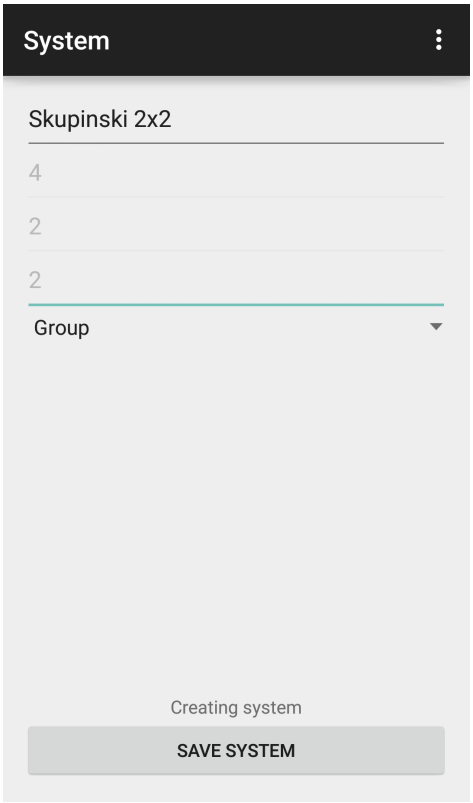
4.6.4 Ustvarjanje novega sistema

Uporabnik lahko ustvari nov sistem s pritiskom na gumb `CREATE` in pravilno izbiro s seznama skupin. Odpre se aktivnost, ki omogoča uporabniku ustvarjanje novega sistema s določenimi nastavitvami. V tekstovna polja se vpiše število tekmovalcev na turnirju, število skupin in število tekmovalcev na skupino. Omenjena polja omogočajo uporabniku, da ustvari tekmovanje po svo-

jih specifikacijah. V spustnem seznamu mora uporabnik izbrati nastavitev, po kateri bo izpeljal tekmovanje. Po izboru nastavitve se opravi preverjanje vnesenih števil. Številke se spremenijo, če se sistem ne more izpeljati in se nadomestijo s primernejšimi številkami, glede na podano število tekmovalcev in izbrani sistem. Uporabnik lahko vedno popravi svoje vnose z izbiro najvišje izbire v spustnem seznamu nastavitev. S pritiskom na gumb `SAVE SYSTEM` se opravi akcija shranjevanja novega sistema v podatkovno strukturo. Uporabnika se obvesti o uspehu akcije shranjevanja.



Slika 4.13: Ustvarjanje sistema.

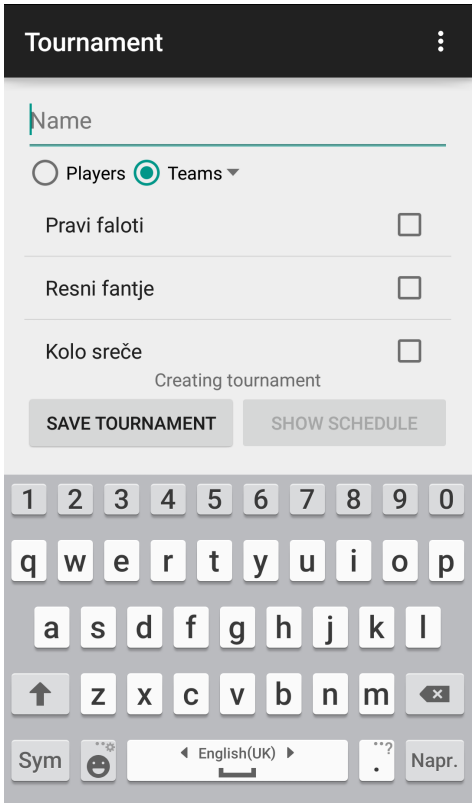


Slika 4.14: Shranjevanje sistema.

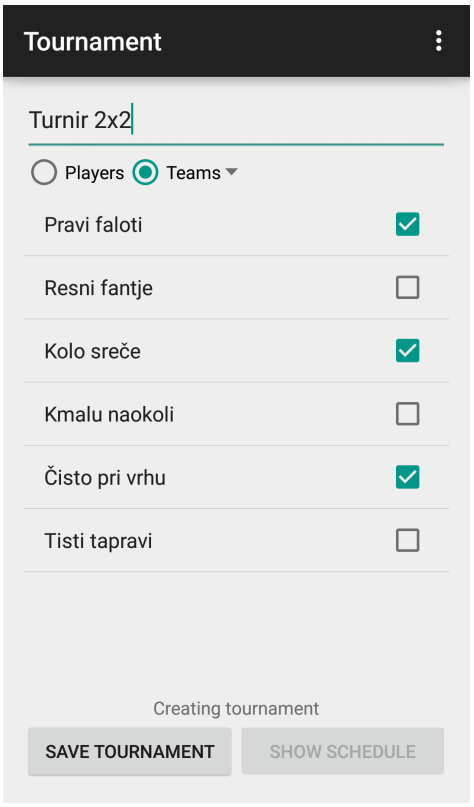
4.6.5 Ustvarjanje novega tekmovanja

Ustvarjanje novega tekmovanja uporabniku omogoča, da vse dosedanje ustvarjene elemente doda v želeno strukturo tekmovanja. Aktivnost omogoča shranjevanje novega tekmovanja samo z imenom, da lahko uporabnik pozneje

ureja nastavitve do konca. Uporabnik lahko na posamezno tekmovanje dodaja in odstrani igralce ali ekipe. Na istem tekmovanju se istočasno ne morejo dodati igralci in ekipe hkrati. Menjava opcij med igralci in ekipami ohranja označene elemente in jih ne odstrani, dokler uporabnik ne ustvari novega tekmovanja ali zapusti aktivnost. Sisteme, ki jih lahko uporabnik izbere s spustnega seznama se prikažejo, ko število izbranih elementov doseže število tekmovalcev, shranjenih v posameznih sistemih.



Slika 4.15: Začetek ustvarjanja tekmovanja.

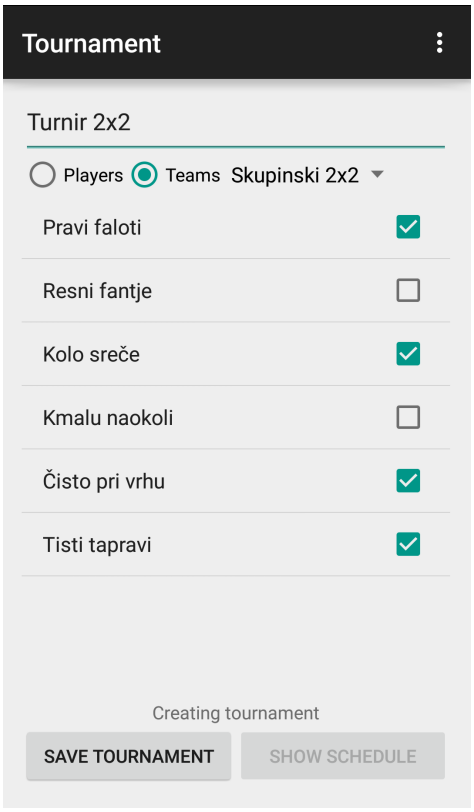


Slika 4.16: Označitev sodelujočih ekip na tekmovanju.

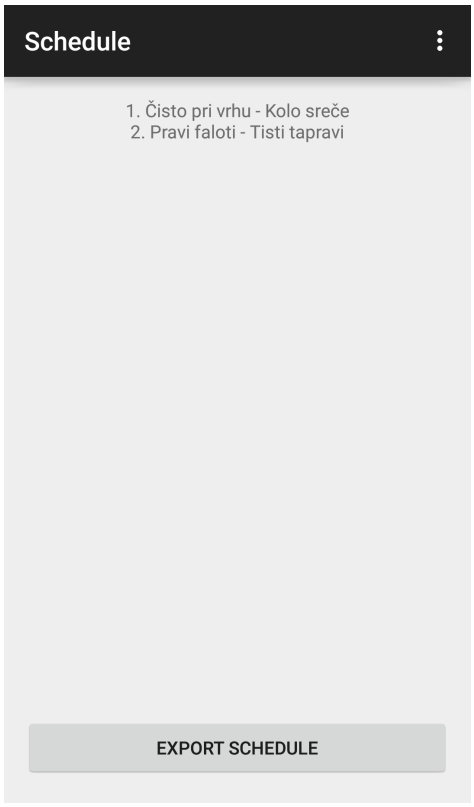
4.6.6 Shranjevanje tekmovanja in pregled žreba

Uporabniku se v spustnemu seznamu prikažejo sistemi, ki zadostujejo njegovim potrebam. Novo tekmovanje lahko uporabnik shrani s pritiskom na

gumb `SAVE TOURNAMENT` in s tem sproži akcijo, ki ustvari nov zapis tekmovanja v podatkovni bazi. Poleg novega zapisa se zapišejo tudi povezave z igralci ali ekipami. Med shranjevanjem se glede na nastavitve opravi žreb tekmovanja, ki se uporabniku prikaže v novi aktivnosti `Schedule`. Uporabnik lahko med urejanjem tekmovanja vedno pogleda nastali žreb. S pritiskom na gumb `EXPORT SCHEDULE` se izvozijo rezultati žreba v tekstovno datoteko, kar omogoča uporabniku lažjo prenosljivost informacij. Žreb se lahko ponovi, če uporabnik ponovno shrani tekmovanje. Vedno obstaja možnost odstranjevanja in dodajanja igralcev ali ekip.



Slika 4.17: Izbira sistema na tekmovanju.



Slika 4.18: Pregled žreba na tekmovanju.

Poglavje 5

Analiza realizacije

Za primerjavo smo si ogledali še podobne rešitve, kot so:

- `Konkuri`,
- `The Tournaments Manager`,
- `Tournament Manager`.

Opravili smo primer žreba na resničnih podatkih, ki smo jih prejeli od Športnega društva Nova Štifta.

5.1 Primerjava z obstoječimi rešitvami

Za primerjavo smo si ogledali nekatere aplikacije, ki se ukvarjajo z organizacijo tekmovanj.

Prvo rešitev smo našli na spletu z imenom `Konkuri` [11]. Omenjena spletna aplikacija omogoča organizacijo tekmovanja glede na športno panogo. Na zagezku uporabe je celotna aplikacija videti zelo uporabna in zagotavlja brezplačno storitev. Po ustvarjanju prvega tekmovanja ugotovimo, da storitev ni povsem brezplačna. Za tekmovanje, na katerem sodelujejo ekipe, je potrebno plačati določen znesek, preden ga lahko aktiviramo. Posamezni ekipi lahko dodajamo igralce, vendar so njihovi podatki dokaj omejeni, če jih

ne povežemo z njihovimi osebnimi računi. Seveda moramo izpostaviti dejstvo, da potrebujemo internetno povezavo, da lahko dostopamo do njihove aplikacije.

Kot naslednjo aplikacijo smo si ogledali mobilno rešitev `The Tournaments Manager` [12], katere opis nam ponuja enostavno in hitro uporabo. Aplikacija je enostavna za uporabo in vizualno privlačna. Poleg tega ponuja žreb tekmovanja, če se odločimo za izločilni sistem. Na žalost nam ne ponuja možnosti skupinskega tekmovanja in ustvarjanja ekip z dodeljevanjem igralcev. Načeloma lahko ustvarimo za vsako skupino novo tekmovanje, vendar ni optimalno.

Na koncu smo si ogledali še `Tournament Manager` [13], katere prednosti so obveščanje uporabnikov. Na žalost aplikacija potrebuje internetno povezavo za ustvarjanje tekmovanj in s tem oteži uporabo na nedostopnih lokacijah. Uporaba je dokaj preprosta, vendar ponuja le dva možna sistema. Za izločitveni sistem lahko opravimo žreb, vendar ne moremo dodeljevati igralcev k ekipam. Poleg tega moramo za vsako tekmovanje posebej ustvarjati tekmovalce, za katere, razen imena, ne moremo podati ostalih podatkov. Mobilni aplikaciji uporabljata reklamne oglase, ki precej motijo med njihovo uporabo.

Naša aplikacija v primerjavi z ostalimi ponuja zastonjsko mobilno rešitev brez oglasnih sporočil. Možnost ustvarjanja lastnih sistemov omogoča organizatorju večjo fleksibilnost in ponovno uporabo. Na priloženih tabelah 5.1 in 5.2 si lahko ogledamo lastnosti posameznih aplikacij v primerjavi z našo aplikacijo `Tournament Coordinator`.

Tabela 5.1: Legenda aplikacij

zap. št.	**aplikacija**
1	Konkuri
2	The Tournaments Manager
3	Tournament Manager
4	Tournament Coordinator

Tabela 5.2: Pregled lastnosti aplikacij

lastnost	**1**	**2**	**3**	**4**
potrebna internetna povezava	da	ne	da	ne
mobilna aplikacija	ne	da	da	da
brezplačna	ne	da	da	da
reklamni oglasi	ne	da	da	ne
različne športne panoge	da	da	da	da
število privzetih sistemov	3	4	2	4
žreb tekmovanja	da	da	da	da

5.2 Primer uporabe

Športno društvo Nova Štifta nam je posredovalo podatke o jesenskemu delu lige v izžrebanih trojkah v nogometu 2015, ki so navedeni v tabeli 5.3. Potrebujejo aplikacijo s katero bi lahko igralce razporedili v ekipe. Na podlagi pridobljenih podatkov smo z uporabo naše aplikacije ustvarili hipotetični naključni žreb in ga primerjali z žrebom našega sistema.

Tabela 5.3: Lestvica lige trojk 2015

igralec	**točke**	**igralec**	**točke**
David Krznar	28	Boštjan Žerovnik	16
Jaka Resnik	28	Matic Remšak	16
Matic Ajnik	26	Mitja Stenšak	14
Niko Sovinšek	26	Tadej Šinkovec	8
Domen Kerznar	20	Luka Šinkovec	8
Štefan Šurk	20	Tomaž Suhovršnik	6

Na tekmovanju v izžrebanih trojkah se ponavadi uporabi naključen žreb igralcev v ekipe, ki nato odigrajo dvoboje, glede na podano shemo. Na sliki 5.4 si lahko ogledamo naključen žreb igralcev v novo nastale ekipe.

Tabela 5.4: Naključen žreb

ekipa 1	**ekipa 2**	**ekipa 3**	**ekipa 4**
Jaka Resnik	Matic Remšak	Luka Šinkovec	Tadej Šinkovec
David Krznar	Matic Ajnik	Boštjan Žerovnik	Domen Kerznar
Štefan Šurk	Mitja Stenšak	Niko Sovinšek	Tomaž Suhovršnik

Na podlagi žreba ugotovimo, da sta se v prvo ekipo uvrstila trenutno najboljša igralca na tekmovanju. Za pravičnejšo razporeditev lahko uporabimo naš sistem. Za primerjavo si oglejmo žreb z uporabo elementov sistema za mali nagomet.

Tabela 5.5: Žreb z novim sistemom

ekipa 1	**ekipa 2**	**ekipa 3**	**ekipa 4**
Niko Sovinšek	David Krznar	Matic Ajnik	Jaka Resnik
Luka Šinkovec	Matic Remšak	Domen Kerznar	Tomaž Suhovršnik
Boštjan Žerovnik	Štefan Šurk	Mitja Stenšak	Tadej Šinkovec

Najboljši igralci na tekmovanju so se naključno razporedili v različne ekipe. Zatem se razporedijo v ekipe še preostali igralci. S tem so se ustvarile pravičnejše ekipe. Število ustvarjenih ekip je premajhno, da bi se ustvarile skupine in naš sistem predlaga potek tekem predstavljen v tabeli 5.6.

Tabela 5.6: Shema tekmovanja

zap. št.	**tekma**	**zap. št.**	**tekma**
1	ekipa1 : ekipa2	4	ekipa2 : ekipa4
2	ekipa3 : ekipa4	5	ekipa1 : ekipa4
3	ekipa1 : ekipa3	6	ekipa2 : ekipa3

Ugotovitve smo posredovali društvu, ki je bilo z delovanjem aplikacije zadovoljno. Aplikacija se bo začela uporabljati že v spomladanskem delu lige izžrebanih trojk.

Poglavje 6

Zaključek

Z rezultatom naše diplomske naloge smo nadvse zadovoljni. Naša aplikacija `Tournament Coordinator` je hitra, enostavna za uporabo ter intuitivna rešitev za vodenje športnih tekmovanj. Je brezplačna rešitev, ki ne potrebuje vzpostavljene internetne povezave. Uporabnik si lahko zamisli svoje parametre za željeni sistem, kar omogoča uporabo na različnih tekmovanjih in različnih športnih panogah. Dodeljevanje igralcev ekipam je hitro, pregled žreba je enostaven in pregleden. Celotna aplikacije je ustvarjena s preprostim vodenjem uporabnika skozi postopek do novega tekmovanja. Aplikacija se lahko uporabi za različne scenarije žrebanja, kot so:

- žrebanje igralcev v ekipe ali skupine,
- žrebanje ekip v skupine,
- žrebanje medsebojnih dvobojev.

Uspelo nam je postaviti temelje za aplikacijo, ki se bo kmalu že uporabila za organizacijske namene.

Med delom na diplomski nalogi nam je uspelo raziskati turnirske razporeditvene sisteme in vključiti funkcionalnosti, ki omogočajo lažjo organizacijo tekmovanj. Ustvarili smo odziven in uporabniku prijazen uporabniški vmesnik, ki omogoča ustvarjanje, urejanje ter pregled igralcev, ekip, sistemov in tekmovanj. Med shranjevanjem novega tekmovanja se opravi žreb, ki se

uporabniku na koncu prikaže. Med razvojem aplikacije so se pojavile ideje za nove funkcionalnosti, kot so:

- zapisovanje tekem,
- objavljanje rezultatov v oblaku,
- opozarjanje uporabnika na približajoče tekme.

Navsezadnje obstaja želja, da bi lahko bili uporabniki obveščeni o dogodkih na tekmovanjih.

Literatura

[1] Turnirski razporeditveni sistemi. [Online]. Dosegljivo: http://www.thefreedictionary.com/tournament. [Dostopano 17. 02. 2015].

[2] Krožni sistem. [Online]. Dosegljivo: http://www.merriam-webster.com/dictionary/round-robin. [Dostopano 17. 02. 2016].

[3] Švicarski sistem. [Online]. Dosegljivo: https://www.fide.com/component/handbook/?id=84&view=article. [Dostopano 17. 02. 2016].

[4] Izločilni sistem. [Online]. Dosegljivo: http://www.curlingcalendar.com/wiki/59. [Dostopano 17. 02. 2016].

[5] Žreb za svetovno prvenstvo v nogometu. [Online]. Dosegljivo: http://www.uefa.com/worldcup/news/newsid=2263977.html. [Dostopano 16. 11. 2015].

[6] Sistem ELO. [Online]. Dosegljivo: http://leagueoflegends.wikia.com/wiki/Elo_rating_system. [Dostopano 16. 11. 2015].

[7] Ronan Schwarz, Phil Dutson, James Steele and Nelson To, *The Android Developer's Cookbook: Building Applications with the Android SDK*, druga izdaja, Developer's Library, 2014.

[8] Belen Cruz Zapata, *Android Studio Essentials*, Packt Publishing, 2015.

[9] Scott Chacon and Ben Straub, *Pro Git*, druga izdaja, Apress, 2014.

[10] Sunny Kumar Aditya and Vikash Kumar Karn, *Android SQLite Essentials*, Packt Publishing, 2014.

[11] Konkuri. [Online]. Dosegljivo: http://www.konkuri.com/ [Dostopano 21. 02. 2016].

[12] The Tournaments Manager. [Online]. Dosegljivo: https://play.google.com/store/apps/details?id=rockets.thetournamentmanager [Dostopano 21. 02. 2016].

[13] Tournament Manager. [Online]. Dosegljivo: https://play.google.com/store/apps/details?id=com.poquesoft.mistorneos [Dostopano 21. 02. 2016].